Model Driven Architecture
A New Paradigm in Software Development

Hermod Opstvedt
Chief Architect
DnB NOR ITU, Norway
Model Driven Architecture

- Model Driven Architecture (MDA) is a specification by the Object Management Group (OMG).
- MDA established as the base architecture for OMG's standards, in September 2001.
- MDA Guide 1.0.1 released in June 2003
- Final version expected late this year
Model Driven Architecture

- OMG's MDA Vision:

  "Support interoperability with specifications that address integration through the entire systems life cycle: from business modeling to system design, to component construction, to assembly, integration, deployment, management, and evolution."
Model Driven Architecture

Why do we need MDA?

➢ We will never all agree on platform specific issues like:
  • Hardware
  • OS
  • Programming language
  • *etc.*
Model Driven Architecture

➢ Need to preserve investments as the computing environment changes.
  • New technologies emerge – need for Integration.
    ✔ What do you do when super technology "BubbA" comes along and your business people just "have-to-have-it".
      ❖ Start a migration project ?
      ❖ Struggle with integration ?
  • New versions – backward incompatibility.
    ✔ Some versions are backward incompatible.
    ✔ New features are added.
Model Driven Architecture

Would it not be nice if you could:

• Install a new add-on from your vendor and just hit the button and have it generate the new stuff?

• Or

• Just change to a different vendor and import your PIM and have it do the generation of this new stuff.
Model Driven Architecture

Software development today

- Traditional project
  - Start with analysis & design.
  - Often paper based diagrams (UML models *etc.*).
  - Sometimes modeling tool is able to generate skeleton code stubs.
  - Move on to coding.
Model Driven Architecture

- More coding – don't have time to update docs now.
- Design documentation gradually becomes out of sync. Even wrong.
- Design documentation is rarely updated in after hand.
Model Driven Architecture

➢ XP Approach

• Coding & Test cases only.
• No documentation – except in code (maybe).
• Project ends and the system moves to the maintenance phase. Original coders leave.
• Maintenance people now have the challenge of trying to understand what the system really does.
• They have no clue as to what was the intention behind it from start.
Model Driven Architecture

- Documentation
  - Often seen as a burden by developers.
  - Can be generated from code in some languages (JavaDoc, etc.).
  - Is a requirement for maintenance.
  - Describes design decisions that can not be expressed in code.
Model Driven Architecture

- How can MDA help us?
- Supports full lifecycle.
  - Analysis.
  - Design.
  - Implementation.
  - Integration
Model Driven Architecture

MDA is based on other standards

- Unified Modeling language (UML).
- Object Constraint Language (OCL).
- XML.
- XML Metadata Interchange (XMI).
- Meta Object Facility (MOF).
- Common Warehouse Metadata.
- etc.
Model Driven Architecture

- Several layers
  - Platform Independent Model (PIM).
  - Platform Specific Model(s) (PSM).
  - Code.
Model Driven Architecture

- Describing a model (PIM)
  - Use some form of language that is suitable for transformation – meaning it can be understood by a computer. Must be well defined.
    - UML
      - Very well suited for the static parts of the model – *i.e.* the class diagrams, *etc.*
    - OCL
      - Very well suited for the dynamic parts of the model – *i.e.* the action/sequence diagrams
Model Driven Architecture

- Describing a model (PIM)
  - The model must describe the business – not the developers’ thoughts of how it should be implemented. (common pitfall)
Model Driven Architecture

Conceptual View

UML based domain (business) model

Your current runtime technology (J2EE, CORBA, .NET, Rel.Database, etc.)

Your language (Java, C/C++/C#, VB.net, SQL, etc.)
Model Driven Architecture

- Platform Independent Model (PIM)
  - Describes the business functionality.
  - Closely matches the user requirements.
  - Not tied to any platform.
  - Can describe pre & post conditions (OCL).
Model Driven Architecture

- Can be mapped into your current environment using a transformation process.
  - A PIM can and normally will be transformed into more than one Platform Specific Model (PSM).
    - J2EE
    - SQL (for database)
    - etc.
Model Driven Architecture

- Use tags or other metadata to annotate the PIM such that the different PSMs can be generated.
- Uses languages derived from the UML language
  - They are based on the Meta Object Facility (MOF)
- Often based on profiles.
  - JSR 26: UML/EJB Mapping Specification
Model Drive Architecture

- **PIM/PSM Transformation**
  - Is the process of taking the PIM into the PSM(s).
  - Can be very complex.
  - Normally done by experts who understand the PSM very well.
  - OMG is working on a standard language
    - Query, Views and Transformation (QVT).
Model Driven Architecture

- Platform Specific Model (PSM)
  - Is the specific representation of the PIM for your target environment.
    - J2EE (WEB + EJB)
    - SQL
    - .NET
  - Is the basis for the next step – Code generation.
Model Drive Architecture

- Platform Specific Model (PSM)
  - Can also be modified in tooling.
    - Dependent of the tool implementation.
  - Can also be reversed into the PIM.
    - Dependent of the tool implementation.
Model Driven Architecture

- Code generation (PSM to Code Transformation)
  - Is the process of taking the PSM into the final destination (code, ddl, *etc.*)
  - This is the least complex part of the process.
Model Driven Architecture

- Code generation (PSM to Code Transformation)
  - Can be developed by anyone that has a good understanding of XMI and the destination code.
  - Can be, and often is, template based.
Model Driven Architecture

- Shortcutting it.
  - Transforming the PIM directly to code
    - Ex. AndroMDA
    - No refining in the PSM stage can be done
Model Driven Architecture

Overview of the complete process.
Model Driven Architecture

- A sample run using opensource software
  - Poseidon for UML – For modeling PIM.
  - AndroMDA – For transforming PIM to code.
Model Driven Architecture

- UML model to code using OpenSource
- A look at the AndroMDA sample
  - The sample model
  - The PIM to code transformation
Model Driven Architecture

Notes about UML modeling.

Because we are using the AndroMDA for generation, there are some requirements we must meet.

- Use the correct Stereotype.
- Use correct types when declaring attributes.
- Primary keys of Entity Beans must be Strings.
Model Driven Architecture

- If EJB's are the target then Session Bean names have to end in the word 'Service'.
- Model one exception class per component.
- See AndroMDA documentation for others.
Model Driven Architecture

- First steps.
  - Run the AndroMDA genapp goal
    - Creates an initial project structure (Eclipse).
    - Creates the necessary Maven artifacts.
  - Model your application (ex. Poseidon for UML).
    - Note the AndroMDA requirements
Model Driven Architecture

- Transforming PIM to code
  - Export the UML model to XMI.
  - Run maven on the generated pom.
Model Driven Architecture

- Is it a paradigm shift?
  - 2-3 years before it is fully mature and adopted (Cutter Consortium).
  - Vendor adoption is increasing (IBM, Compuware, \textit{etc.})
  - Immature tooling.
Model Driven Architecture

- Prediction is that it will have 50% market share by 2007 in the OO tools market (Cutter Consortium).
- Some vendors have documented 30% decrease in time spent on project.
Model Driven Architecture

- Conclusion:
  - MODEL – Don't CODE !!
Model Driven Architecture

Some tools available:

- Opensource
  - AndroMDA: http://www.andromda.org/
  - Jamda: http://jamda.sourceforge.net
Model Driven Architecture

Commercial

- Compuware: OptimalJ (Netbeans based)
- IBM: Rational XDE (Eclipse based)
- innoQ: iQgen
- Io-Software: ArcStyler
Model Driven Architecture

- Resources:
  - [http://www.omg.org/mda](http://www.omg.org/mda)
Model Driven Architecture

- Questions